STQA-Session 9

Black Box Testing (3)

Decision Table

章晓芳

xfzhang@suda.edu.cn

Black Box Testing Techniques

- Boundary Value Analysis
- Equivalence Partitioning
- Decision Table
- Cause-Effect Graph
- Combinatorial Test

Black Box Testing Techniques

- Review
  - Boundary Value Analysis
  - Equivalence Partitioning

Decision Table

A black-box test design technique in which test cases are designed to execute the combinations of inputs.

Decision table is good for representing and analyzing complex logical relationships.

Decision Table Template

Decision Table

Limited entry decision table: All the conditions are binary

Extended entry decision table:Conditions are allowed to have several values

Mixed entry decision table

Limited Entry Decision Tables

Don't care: the condition is irrelevant, or the condition does not apply

	Rules	1	2	3-4	5	6	7-8
Condition	c1	T	T	T	F	F	F
	c2	T	T	F	T	T	F
	c3	T	F	—	T	F	—
Action	a1	X	X		X		
	a2	X				X	
	a3		X		X		
	a4			X			X

Condition: input/equivalence class of input

Action: output/major functional processing portions of the item tested

Extended Entry Decision Tables

	Rules	1	2	3
Condition	c1	0-10	0-10	10-100
	c2	<5	5	>5
Action	a1	DoX	DoX	DoY
	a2	DoA	DoB	DoC

Mixed Entry Decision Tables

Cutting Machines		R1	R2	R3	R4	R5	R6	R7	R8
C1	Turning diameter	D ≤ 200	200 < D ≤ 500	D ≤ 200	200 < D ≤ 500	D ≤ 200	200 < D ≤ 500	D ≤ 200	200 < D ≤ 500
C2	Turning length	L ≤ 1000	L ≤ 1000	1000 < L ≤ 1500	1000 < L ≤ 1500	L ≤ 1000	L ≤ 1000	1000 < L ≤ 1500	1000 < L ≤ 1500
C3	Milling needed	N	N	N	N	Y	Y	Y	Y
A1	Machine 4711	X	-	X	-	-	-	-	-
A2	Machine 4712	-	X	-	X	-	-	-	-
A3	Machine 4713	-	-	-	-	-	-	X	X
A4	Machine 4714	-	-	-	-	X	X	-	-

How to Make a Decision Table?

1. List all condition stubs and action stubs
2. Fill the condition items.
3. Fill the action items, make a initial table
4. Decision Table Verification

Decision Table Verification

- Completeness
- Combining
- Redundancy
- Inconsistence

Completeness of decision table

The decision table should contain every combination of Predicate values

For limited entry decision tables, if n conditions exist, there must be 2^n rules.

When *don't care* entries really indicate that the condition is irrelevant,we can develop a rule count as follows:

- rules in which no don't care entries occur count as one rule
- each don't care entry in a rule doubles the count of that rule.

What about extended entry decision tables?

Combining rules in Decision table

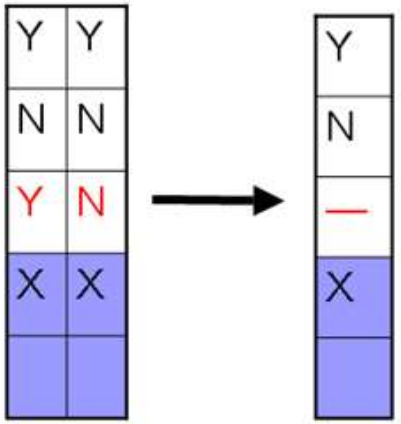
Consider the following decision table:

Rules	1	2	3	4	5	6	7	8
c1	T	T	T	T	F	F	F	F
c2	T	T	F	F	T	T	F	F
c3	T	F	T	F	T	F	T	F
a1			X	X			X	X
a2	X	X			X			
a3			X	X			X	X
a4						X		

Multiple rules may be combined when multiple rules with the same condition and same outcome

similar

Which rule(s) may be combined?



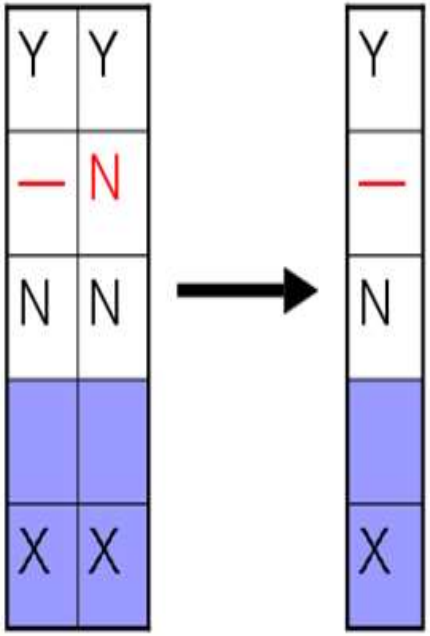
Rules	1	2	3	4	5	6	7	8
c1	T	T	T	T	F	F	F	F
c2	T	T	F	F	T	T	F	F
c3	T	F	T	F	T	F	T	F
a1			X	X			X	X
a2	X	X			X			
a3			X	X			X	X
a4						X		

Rules	1	2	3-4	5	6	7-8
c1	T	T	T	F	F	F
c2	T	T	F	T	T	F
c3	T	F	-	T	F	-
a1			X			X
a2	X	X		X		
a3			X			X
a4					X	

Check completeness

Rules	1-2	3-4, 7-8	5	6
c1	T	-	F	F
c2	T	F	T	T
c3	-	-	T	F
a1		X		
a2	X		X	
a3		X		
a4				X

Other Contracted Decision Table?



Redundant Decision Table

tub	1–4	5	6	7	8	9
c1	T	F	F	F	F	T
c2	—	T	T	F	F	F
c3	—	T	F	T	F	F
a1	X	X	X	—	—	X
a2	—	X	X	X	—	—
a3	X	—	X	X	X	X

What is the redundancy in this decision table?

Inconsistent Decision Table

Every combination of predicate truth values results in only one action or set of actions

Stub	1–4	5	6	7	8	9
c1	T	F	F	F	F	T
c2	—	T	T	F	F	F
c3	—	T	F	T	F	F
a1	X	X	X	—	—	—
a2	—	X	X	X	—	X
a3	X	—	X	X	X	—

What is the problem of this decision table?

How to Make a Decision Table?

1. List all condition stubs and action stubs
2. Fill the condition items.
3. Fill the action items, make a initial table
4. Decision Table Verification
   - Completeness
   - Combining
   - Redundancy
   - Inconsistence

Examples

- Triangle problem
- NextDate problem

Decision Table for the Triangle Problem

How many rules?

Don't care entries

Based on EP results

Stub	1	2	3	4	5	6	7	8	9
c1: a, b, c form a triangle?	F	T	T	T	T	T	T	T	T
c2: a = b?	—	T	T	T	T	F	F	F	F
c3: a = c?	—	T	T	F	F	T	T	F	F
c4: b = c?	—	T	F	T	F	T	F	T	F
a1: Not a Triangle	X								
a2: Scalene									X
a3: Isosceles					X		X	X	
a4: Equilateral		X							
a5: Impossible			X	X		X			

Impossible rule

Infeasible equivalence class

Decision Table for the Triangle Problem (Refined version with rule count)

Stub	1	2	3	4	5	6	7	8	9	10	11
c1: a < b + c?	F	T	T	T	T	T	T	T	T	T	T
c2: b < a + c?	—	F	T	T	T	T	T	T	T	T	T
c3: c < a + b?	—	—	F	T	T	T	T	T	T	T	T
c4: a = b?	—	—	—	T	T	T	T	F	F	F	F
c5: a = c?	—	—	—	T	T	F	F	T	T	F	F
c6: b = c?	—	—	—	T	F	T	F	T	F	T	F
Rule Count	32	16	8	1	1	1	1	1	1	1	1
a1: Not a Triangle	X	X	X								
a2: Scalene											X
a3: Isosceles							X		X	X	
a4: Equilateral				X							
a5: Impossible					X	X		X			

Test Cases for the triangle problem

Case ID	a	b	c	Expected Output
DT1	4	1	2	Not a Triangle
DT2	1	4	2	Not a Triangle
DT3	1	2	4	Not a Triangle
DT4	5	5	5	Equilateral
DT5	?	?	?	Impossible
DT6	?	?	?	Impossible
DT7	2	2	3	Isosceles
DT8	?	?	?	Impossible
DT9	2	3	2	Isosceles
DT10	3	2	2	Isosceles
DT11	3	4	5	Scalene

DT5, DT6, DT8 are infeasible ECs

一个简单的练习

- 某公司的对客户分类标准如下：
  - 顾客每次订货额在1000元以上（含1000元），信誉好的，订单设“优先”标志；信誉不好，但是老客户的，订单设“优先”标志；信誉不好，但是新客户的，订单设“正常”标志；
  - 每次订货额在1000元以下，订单设“正常”标志。
- 请绘制相应的决策表。

一个简单的练习

C1:订货额>=1000?
C2: 信誉好？
C3: 老客户？
rules
A1: 订单设“优先”标志
A2: 订单设“正常”标志

一个简单的练习

	1	2	3	4
C1:订货额>=1000?	T	T	T	F
C2: 信誉好？	T	F	F	-
C3: 老客户？	-	T	F	-
# rules	2	1	1	4
A1: 订单设“优先”标志	X	X		
A2: 订单设“正常”标志			X	X

NextDate Input Equivalence Classes

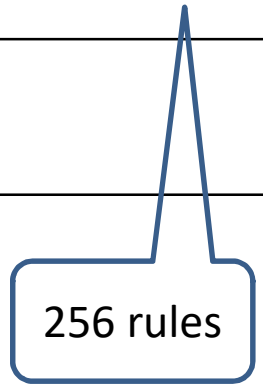
- M1 = { month: month has 30 days}
- M2 = { month: month has 31 days}
- M3 = { month: month is February}
- D1 = {day: 1<= day <= 28}
- D2={day: day=29}
- D3={day: day=30}
- D4={day: day=31}
- Y1={year: leap year};
- Y2={year: not leap year}.

NextDate Output Equivalence Classes

- A1: impossible day
- A2: incremented day value
- A3: date reset
- A4: incremented month value
- A5: month reset
- A6: incremented year value

Limited Entry Decision Table

Condition				
C1: Is month in M1? C2: Is month in M2? C3: Is month in M3? C4: Is date in D1? C5: Is date in D2? C6: Is date in D3? C7: Is date in D4? C8: is year in Y1?	T	T	T	
A1: possible A2: NextDate				



Second trying: extended entry decision table + considering leap year 2000

- Equivalent class set

M1={Month: 30 days}

M2={Month: 31 days}

M3={Month: February}

D1={Date: 1<=Date<=28}

D2={Date: Date=29}

D3={Date: Date=30}

D4={Date: Date=31}

Y1={Year: year=2000}

Y2={Year: is leap year and !=2000}

Y3={Year: is not leap year}

Second trying

	1	2	3	4	5	6	7	8
C1: month	M1	M1	M1	M1	M2	M2	M2	M2
C2: date	D1	D2	D3	D4	D1	D2	D3	D4
C3: year	-	-	-	-	-	-	-	-
# rules	3	3	3	3	3	3	3	3
A1:impossible				X				
A2:date++	X	X			X	X	X	
A3:date reset			X					X
A4:month++			X					?
A5:month reset								?
A6:year++								?

Extended entry decision table

3*4*3=36rules

Second trying (continued)

	9	10	11	12	13	15	15	16
C1: month	M3	M3	M3	M3	M3	M3	M3	M3
C2: date	D1	D1	D1	D2	D2	D2	D3	D4
C3: year	Y1	Y2	Y3	Y1	Y2	Y3	-	-
# rules	1	1	1	1	1	1	3	3
A1:impossible						X	X	X
A2:date++	X	X						
A3:date reset			X	X	X			
A4:month++			X	X	X			
A5:month reset								
A6:year++								

Third trying: consider date and month

- Equivalent class set

M1={Month: 30 days}

M2={Month: 31 days, except December}

M3={Month: December}

M4={Month: February}

D1={Date: 1<=Date<=27}

D2= {Date: Date=28}

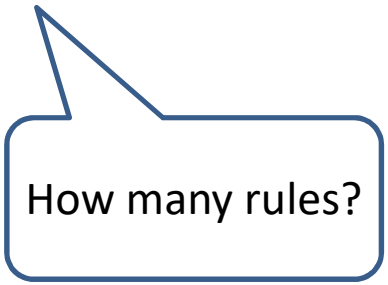
D3={Date: Date=29}

D4={Date: Date=30}

D5={Date: Date=31}

Y1={Year: is leap year}

Y2={Year: is not leap year}



Decision Table for NextDate

	1	2	3	4	5	6	7	8	9	10
c1：月在	M1	M1	M1	M1	M1	M2	M2	M2	M2	M2
c2：日在	D1	D2	D3	D4	D5	D1	D2	D3	D4	D5
c3：年在	—	—	—	—	—	—	—	—	—	—
动作										
a1：不可能					X					
a2：日增 1	X	X	X			X	X	X	X	
a3：日复位				X						X
a4：月增 1				X						X
a5：月复位										
a6：年增 1										

Decision Table for NextDate (continued)

	11	12	13	14	15	16	17	18	19	20	21	22
c1：月在	M3	M3	M3	M3	M3	M4	M4	M4	M4	M4	M4	M4
c2：日在	D1	D2	D3	D4	D5	D1	D2	D2	D3	D3	D4	D5
C3：年在	–				–	–	Y1	Y2	Y1	Y2	–	–
动作												
a1：不可能										X	X	X
a2：日增1	X	X	X	X		X	X					
a3：日复位					X			X	X			
a4：月增1								X	X			
a5：月复位					X							
a6：年增1					X							

Simplified Decision Table for NextDate

	1－3	4	5	6－9	10
c1：月份在	M1	M1	M1	M2	M2
c2：日期在	D1，D2，D3	D4	D5	D1，D2，D3，D4	D5
c3：年在	－	－	－	－	－
动作					
a1：不可能			X		
a2：日期增1	X			X	
a3：日期复位		X			X
a4：月份增1		X			X
a5：月份复位					
a6：年增1					

Simplified Decision Table for NextDate (continued)

	11－14	15	16	17	18	19	20	21－22
c1：月份在	M3	M3	M4	M4	M4	M4	M4	M4
c2：日期在	D1，D2，D3，D4	D5	D1	D2	D2	D3	D3	D4，D5
C3：年在	－	－	－	Y1	Y2	Y1	Y2	－
动作								
a1：不可能							X	X
a2：日期增1	X		X	X				
a3：日期复位		X			X	X		
a4：月份增1					X	X		
a5：月份复位		X						
a6：年增1		X						

13 Test cases of NextDate

用例 ID	月份	日期	年	预期输出
1－3	4	15	2001	2001 年 4 月 16 日
4	4	30	2001	2001 年 5 月 1 日
5	4	31	2001	不可能
6－9	1	15	2001	2001 年 1 月 16 日
10	1	31	2001	2001 年 2 月 1 日
11－14	12	15	2001	2001 年 12 月 16 日
15	12	31	2001	2002 年 1 月 1 日
16	2	15	2001	2001 年 2 月 16 日
17	2	28	2004	2004 年 2 月 29 日
18	2	28	2001	2001 年 3 月 1 日
19	2	29	2004	2004 年 3 月 1 日
20	2	29	2001	不可能
21－22	2	30	2001	不可能

Exercise (consider from another point)

- Equivalent class set

D1={Date: 1<=Date<=27}

D2= {Date: Date=28}

D3={Date: Date=29}

D4={Date: Date=30}

D5={Date: Date=31}

M1={Month: 30 days}

M2={Month: 31 days, except December}

M3={Month: December}

M4={Month: February}

Y1={Year: is leap year}

Y2={Year: is not leap year}

13 Test cases of NextDate

	1	2	3	4	5	6	7	8	9	10	11	12	13
C1:date	D1	D2	D2	D2	D3	D3	D3	D4	D4	D4	D5	D5	D5
C2:month	-	M1,M2, M3	M4	M4	M1,M2, M3	M4	M4	M1	M2,M3	M4	M1,M4	M2	M3
C3:year	-	-	Y1	Y2	-	Y1	Y2	-	-	-	-	-	-
A1: impossible							X			X	X		
A2:date++	X	X	X		X				X				
A3:date reset				X		X		X				X	X
A4:month++				X		X		X				X	
A5:month reset													X
A6:year++													X

BVA VS. EP VS. DT

- 边界值分析

  基于取值域，不识别数据或逻辑关系；

  很容易自动化实现，设计工作量小；

  生成的测试用例数比较多，测试用例执行时间长。

- 等价类技术

  考虑数据依赖关系，标识等价类时需要更多的判断和技巧；

  等价类标识出以后的处理也是机械的；

  设计工作量和测试用例数属中等。

- 决策表技术

  考虑数据的逻辑依赖关系；

  所得测试用例可以是完备的，测试数量在一定意义上讲是最少的；

  需要通过多次迭代，设计工作量很大。

Black Box Testing Techniques

- Boundary Value Analysis
- Equivalence Partitioning
- Decision Table
- Cause-Effect Graph
- Combinatorial Test